基於 AES 對稱加密的 ICMP 隧道通訊下的肉雞群控與 CC 攻擊系統

[redacted]
資訊工程學系
國立高雄大學
中華民國高雄市
[redacted]o365.nuk.edu.tw

摘要—本研究實作一套基於 AES 對稱加密 [2] 與 ICMP 隧道通訊 [1], [4] 的被控端（肉雞）監控與 HTTP Flood（CC 攻擊）系統，旨在於無開放連接埠的環境下，建立具高隱蔽性且安全的遠端控制通道。系統核心利用 ICMP Echo Request 夾帶控制指令，採用 AES-256-CTR 加密模式並搭配每次隨機生成的 16 bytes 初始化向量（IV），確保同一指令可產生不同密文，以規避流量特徵分析並提升抗偵測能力。

架構上，被控端常駐 ICMP 監聽程式進行解密驗證與指令執行，並定期回傳心跳包（Heartbeat）以回報存活狀態；主控端則以 PHP 與 MariaDB 建置 Web 管理介面，負責設備監控（IP、AES Key、最後連線時間）與指令發送。在攻擊功能方面，系統整合 Bombardier 壓力測試工具，支援自定義 HTTP 方法（GET/POST）、Headers、Body、Timeout、併發數與攻擊時長等參數，可自動生成並執行 CC 攻擊指令。實驗結果證實，本系統能在封閉網路環境中穩定維持加密控制通路並有效執行壓力測試，具備網路安全攻防與惡意程式分析之教育研究價值。

關鍵字—CC 攻擊、ICMP 通訊、AES 對稱加密、遠控、肉雞、群控、壓力測試、網路安全

I. 相關研究與技術背景

本研究整合 ICMP 隧道通訊、AES 加密、被控端管理與 CC 壓力測試工具，形成一套可在封閉環境中運作的遠端控制與攻擊模擬系統，在技術整合與教育研究上皆具重要價值。

A. ICMP 協定與隧道化技術

ICMP（Internet Control Message Protocol）主要用於回報網路狀態，例如 Echo Request/Echo Reply [1]。由於其流量在多數環境中不易被封鎖，因此常成為隱蔽通訊研究的基礎 [4], [5]。ICMP 隧道化（ICMP Tunneling）係指將任意資料封裝於 ICMP payload 中，用以在未開放 TCP/UDP 之環境下建立控制通道，常見於惡意軟體 C2 通訊研究 [6]。

B. AES 對稱式加密

AES（Advanced Encryption Standard）[2] 為目前廣泛採用的對稱式加密演算法，具有高效能、強安全性與硬體支援完善等特點。CTR（Counter）模式屬串流加密方式，每次加密均依賴一組隨機初始化向量（IV），使相同明文在不同時間加密後產生不同密文，增加抗分析能力 [2]。

本研究選用 AES-256-CTR 作為 ICMP 隧道的內容保護機制，使控制指令在傳輸過程中無法被第三方直接解讀或還原。

C. HTTP Flood 攻擊原理與壓力測試工具

HTTP Flood 攻擊（CC 攻擊）屬於應用層（Layer 7）的分散式阻斷服務攻擊，其目的為以大量 HTTP 請求耗盡伺服器資源 [7]。本研究整合 Bombardier [8] 作為壓力測試與攻擊模擬工具。其研究價值可分為：

1) 分析不同併發量與請求模式對伺服器的負載；
2) 觀察防禦策略面對大量 HTTP 流量時的反應；
3) 模擬攻擊流量以測試服務韌性；
4) 建立主控端下令、被控端執行的完整攻擊模式。

D. PHP 與 *MariaDB*

本研究後端採用 PHP + MariaDB 作為主控端管理面板的實作基礎。PHP 原生支援調用底層 Raw Socket 工具，使主控端能直接對被控端下達 AES 加

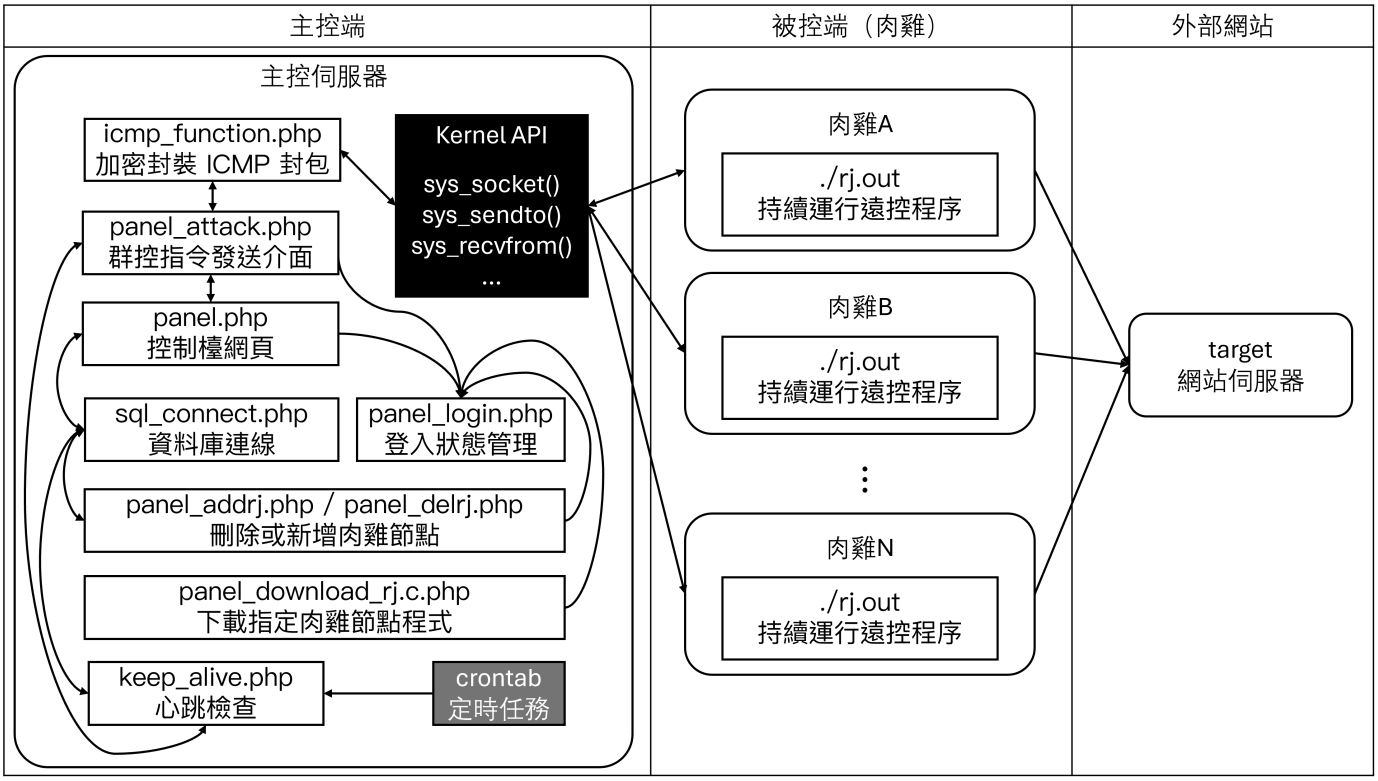

图 1. 完整 ICMP 隧道與 AES 加密資料流程

密的 ICMP 指令。MariaDB 具有高效能，適合作為中小型系統的資料儲存引擎。PHP 與 MariaDB 整合後，可即時更新被控端的最後心跳時間、在線狀態、刪除標記等資料，提供穩定的被控端監控界面。

II. 系統設計

本系統由「主控端」與「被控端」兩大部分組成，如圖 1 所示，其中主控端包含 Web 控制介面、AES 金鑰生成、資料庫管理與 ICMP 封包下發功能；被控端則為常駐程式，負責接收、解密及執行指令並 echo 回應。

A. 主控端

主控端系統依功能拆分為以下模組：

(a) **Key Generator 模組：**產生 AES-256 金鑰並嵌入被控端程式；

(b) **Bot Code Generator 模組：**自動生成可編譯之 C 控端端程式代碼；

(c) **ICMP Control 模組：**組裝加密封包並以 Raw Socket 組裝並發送自訂 ICMP Echo Request，遵循 Unix Network Programming 中所述之封包構造流程 [3]。

(d) **Heartbeat Monitor 模組：**定時廣播心跳並標記離線被控端；

(e) **Command Dispatcher 模組：**下發一般指令與 CC 攻擊指令；

(f) **Database Management 模組：**維護被控端資訊與狀態。

B. 被控端

被控端以 C 語言實作，每個被控端具備獨立 AES 密鑰，提高安全性與被控端隔離性。被控端功能：

(a) 持續監聽 ICMP Echo Request；

(b) 使用 AES-256-CTR 解密 payload；

(c) 執行指令（包含 CC 攻擊指令、心跳包回覆及其他指令）；

(d) 回傳 AES 加密之 ICMP Echo Reply；

C. 系統運作流程

本系統之整體運作流程如下：

1) **新增被控端：**管理者於 Web Panel 輸入目標 IP，由前端 JavaScript 產生 AES 金鑰，後端 PHP 程式自動生成被控端 C 程式碼，並將被控端 IP 與 AES 金鑰寫入資料庫。
2) **心跳廣播：**主控端每 60 秒依序向所有被控端發送加密之 ICMP 心跳包。
3) **被控端回應：**被控端回傳 AES 加密的 ICMP Echo Reply，主控端據此更新資料庫中的最後存活時間。
4) **離線判定：**若被控端於 5 分鐘內未傳回任何回應，則將其標記為 Offline。
5) **群控指令下發：**管理者於 Web Panel 輸入指令後，主控端以加密 ICMP 將指令廣播至所有活躍（非離線）被控端。
6) **指令執行回傳：**被控端解密指令後執行相應操作，並以加密回應封包回傳執行結果。

III. 系統實作細節

A. *ICMP* 加密封包建構與傳輸方式

主控端使用 Raw Socket 主動組裝並發送自訂之 ICMP Echo Request。流程包含 AES 加密、ICMP 封包組裝、校驗和計算與 Raw Socket 傳輸，詳述如下。

1) AES 加密與 *Payload* 建構*:* 主控端以 AES-256-CTR 模式加密指令明文，採用 OpenSSL 之實作，可確保加密流程的正確性與一致性 [9]。加密流程如下：

a) 產生 16 bytes 初始化隨機向量（IV）；
b) 對明文執行 AES-256-CTR 加密；
c) 將「IV || 密文」以 Base64 編碼；
d) 加上 null terminator 成為最終 payload。

此格式確保被控端能正確解析並復原明文。

2) ICMP 封包組裝與校驗和計算*:* ICMP Echo Request 之標頭包含 type、code、checksum、identifier 與 sequence number。本研究流程如下：

a) 建立不含校驗和的暫存封包；
b) 將封包依 16-bit 單位加總後取反，得出 checksum；
c) 重新組合完整 ICMP header；
d) 將 header 與 payload 合併為最終封包。

此步驟確保封包符合 RFC 792 標準。

3) Raw Socket 傳輸加密 *ICMP* 封包*:* 主控端透過 Raw Socket 建立低階傳輸通道，使系統能直接構造並發送自訂的 ICMP 封包。Raw Socket 允許應用程式自行填入 ICMP 標頭、校驗和與加密後的 payload，使封包內容完全由主控端掌控，而不依賴作業系統自動生成。

在封包構造完成後，主控端將加密 ICMP Echo Request 直接送往指定被控端，並於傳輸後進入等待階段，用以接收被控端回傳的 ICMP Echo Reply。此方式使 PHP 能以使用者層程式直接操作網路層（Layer 3），達成自訂封包格式、隧道化控制訊息與加密內容傳遞等功能。

透過 Raw Socket，主控端可精準掌握 ICMP 控制流量，並與被控端形成安全且可管理的加密隧道通道。

4) 回覆等待與超時機制*:* 封包送出後，主控端最多等待 3 秒，若未收到被控端回覆，則視為超時或離線，並更新資料庫中被控端狀態。

IV. 系統缺點

本研究透過 Raw Socket 發送經 AES-CTR 加密的 ICMP Echo Request，被控端解密 payload 並回傳相同加密 Echo Reply。AES-CTR 模式雖具有高效能與良好隱蔽性，但其安全性依賴初始化向量（IV）的不可重複性。若在高頻率或大量被控端環境中意外出現 IV 重複，削弱加密強度，使密文可能被分析。

在一般網路環境中，ICMP Echo Request 主要用於連線檢測，其 payload 通常為固定或小量資料。本研究將 AES 加密後之指令與隨機 IV 封裝於 payload，因此其長度相較於一般 ICMP 封包更大且具有變動性。在多數一般網路環境中，此差異不會造成明顯問題，但若網路部署有進階入侵偵測系統（IDS），有可能因封包大小或傳輸模式異常而被識別。

V. 結論

本研究成功實作一套基於 AES 加密與 ICMP 隧道通訊之遠端控制系統，並建構整合 Web 介面、資料庫管理、心跳監控、群控指令下發及 CC 壓力測試的完整控制框架。透過 Raw Socket 實現自訂 ICMP 封包，使主控端得以在未開放傳統 TCP/UDP 埠的情況下，與被控端建立隱蔽且安全的控制通道。

整體測試結果顯示，本系統具備以下優點：(1) ICMP 加密隧道運作穩定、延遲低；(2) AES-256-CTR

加密模式確保封包內容不易被攔截或解讀；(3) 自動生成被控端程式碼提升部署效率與被控端隔離性；(4) 心跳監控機制可即時掌握被控端狀態；(5) 群控指令及 CC 壓力測試模組可有效模擬分散式控制行為。

本系統成功展示隧道化通訊、安全加密與分散式控制技術之整合應用，也為未來隱蔽通訊、分散式控制系統與惡意行為分析等領域奠定了良好基礎。

參考文獻

[1] J. Postel, “Internet Control Message Protocol,” RFC 792, IETF, 1981.

[2] NIST, “Announcing the Advanced Encryption Standard (AES),” FIPS PUB 197, 2001.

[3] W. R. Stevens, “Unix Network Programming, Volume 1: The Sockets Networking API,” 3rd ed., Addison-Wesley, 2003.

[4] L. P. Cox, M. C. Chan, and V. N. Padmanabhan, “Practical ICMP-based IP mobility,” in *Proc. IEEE INFOCOM*, 2004.

[5] S. Zander, G. Armitage, and P. Branch, “A survey of covert channels and countermeasures in computer network protocols,” *IEEE Communications Surveys & Tutorials*, 2007.

[6] M. Bailey et al., “The underground economy of botnets,” in *Proc. Workshop on Economics of Information Security*, 2007.

[7] S. T. Zargar, J. Joshi, and D. Tipper, “A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013.

[8] A. Makhov, “Bombardier: Fast cross-platform HTTP benchmarking tool,” GitHub, 2020. [Online]. Available: https://github.com/codesenberg/bombardier

[9] B. Damgård and I. Damgård, “Understanding cryptography in web applications: secure use of OpenSSL,” Springer, 2011.

[10] C. Benvenuti, *Understanding Linux Network Internals*. Sebastopol, CA, USA: O’Reilly Media, 2005, pp. 600–615.